

SDK for APS Master Interface[®]

Version 2.0

User Manual

APS[®], Automatic Payment Systems
April 2009



Revision Sheet

Release No.	Date	Revision Description
1.0	Feb 3rd, 2009	Initial Release

1.0 GENERAL INFORMATION

1.1 System Overview

The APS Master Interface[®] allows connecting a Bill Validator and/or Coin Acceptor using the standard MDB Vending Protocol to a PC running Microsoft Windows XP. The APS Master Interface[®] is connected to one of the USB ports in the PC.

A Multidrop Bus (MDB) is a computer bus in which all components are connected to the same set of electrical wires. A process of arbitration determines which device gets the right to be the sender of information at any point in time. The other devices must listen for the data that is intended to be received by them.

Multidrop Buses are used by Vending Machine controllers to communicate with the vending machine's components. The APS Master Interface[®] opens the doors for a wide range of customized applications using a simple PC to control all the machine's components.

The Software Development Kit (SDK) for the APS Master Interface[®] includes all the drivers required to operate the equipment and also a Visual Basic sample code to speed up the learning process on creating new applications with this system.

1.2 Authorized Use Permission

Usage of the APS Master Interface[®] software is limited to its owner via the APS Terms and Conditions (26/08/2004).

1.3 Points of Contact

For additional information, APS Team can be contacted through our technical support email at techsupport@apsmx.com

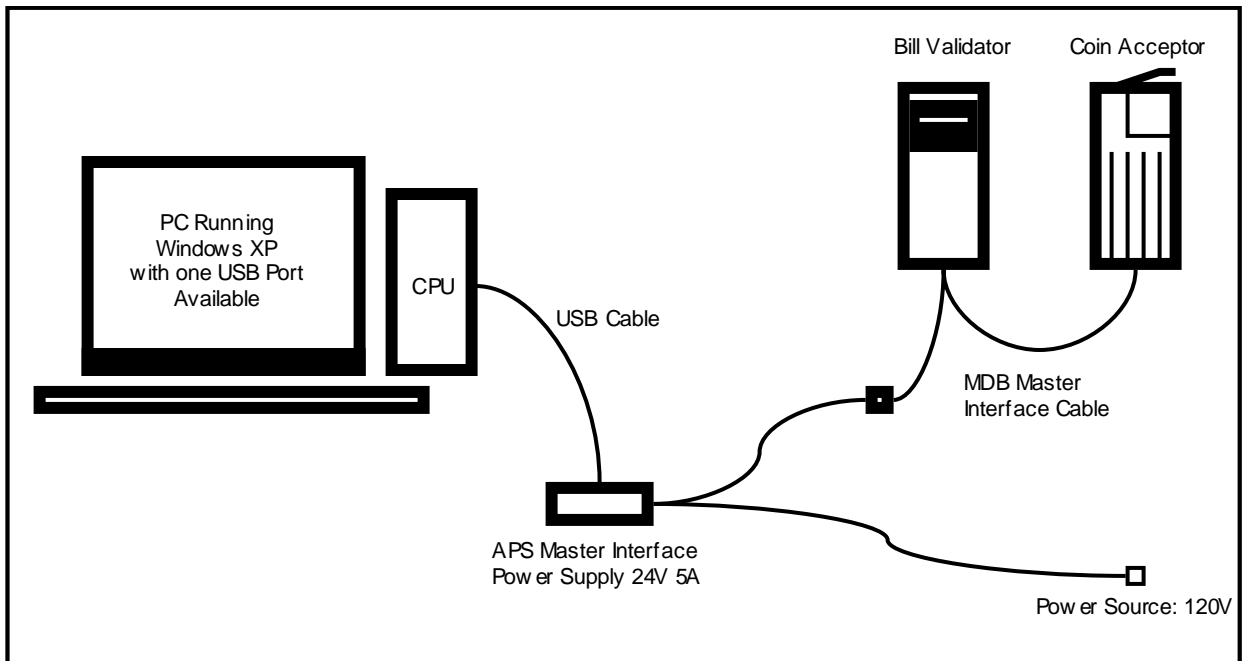
1.4 Acronyms and Abbreviations

MDB	Multidrop Bus
SDK	Software Development Kit
APS	Automatic Payment Systems
AMI	APS Master Interface

2.0 SYSTEM SUMMARY

2.1 Hardware Configuration

The APS Master Interface[®] (AMI) is configured to be connected to one available USB port in a PC Running Windows XP. Using the MDB Master Interface Cable it is possible to connect a Bill Validator and/or a Coin Acceptor.



AMI is powered up by the USB connection. An additional External Power Supply is required to power up the Bill Validator and/or Coin Acceptor.

AMI contains the electrical interface for MDB master, the device is compliant with the timing of the MDB 2.0 Specification. Some low level parts of the MDB communication is implemented directly on the interface microcontroller. This allow to meet the MDB timing specification and to control the mode bit of the MDB protocol.

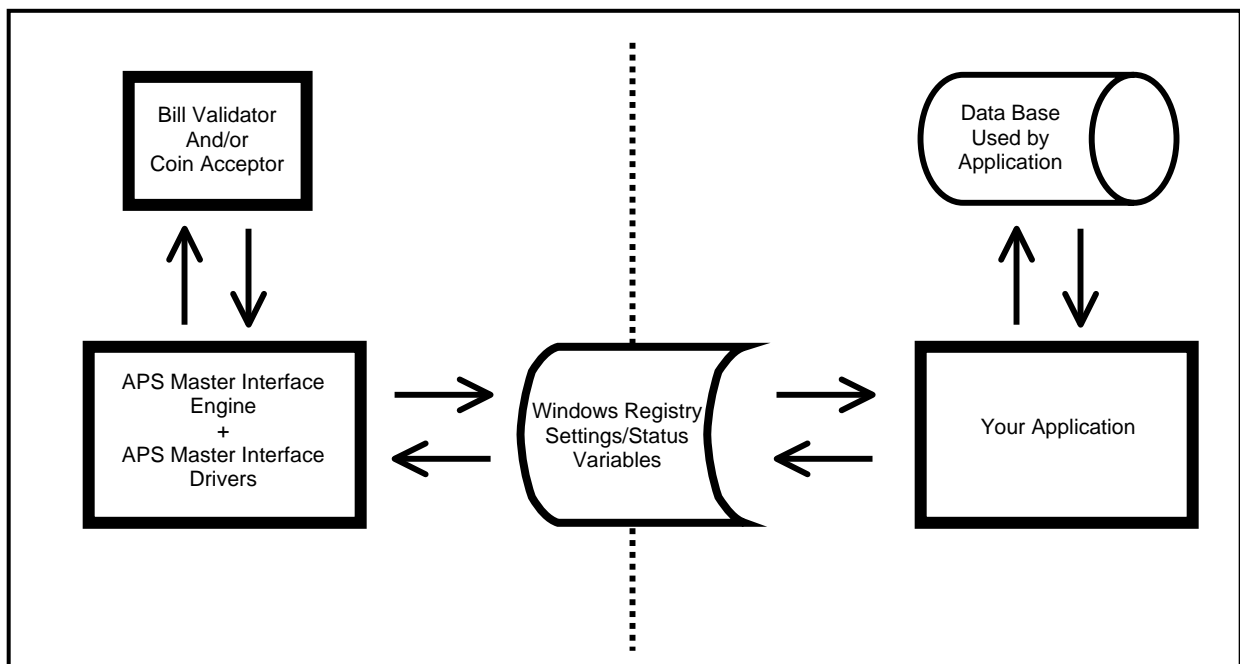
To communicate with a MDB slave, we must use the MDB master of the interface. The timing and mode bit control is handled by the interface microcontroller. The full MDB Data Frame, including command and checksum, is send to the interface endpoint. The response of the MDB slave is then read from the interface endpoint.

2.2 Software Configuration

APS Master Interface Software was developed to allow the interaction of any application written in any programming language that is able to communicate and change "Windows Registry" values in the Windows XP Operative System.

AMI Software consists on two different modules:

- a) APS Master Interface Drivers: Used to control the AMI Hardware via USB Port.
- b) APS Master Interface Engine: Used to control the functions of the AMI Hardware including the operation of one Bill Validator and/or one Coin Acceptor. The AMI Engine is also the link that communicates those devices with Your Application using Windows Registry variables/values.



Windows Registry values are used to store AMI Engine settings such as General Settings for Bill Validator and Coin Acceptor, type of bills to accept, value of the type of bills accepted, type of coins to accept, enable/disable delivery of change, etc.

The AMI Engine must be initialized before the payment procedure starts. The AMI Engine will run in the background waiting for the Application to trigger the payment process.

A typical Application consist on a software that request the user to provide an account number or a selection of a product, then the system connects to a Data Base to get the amount of money that the user will pay using the Bill Validator and/or Coin Acceptor.

Once the amount of money to be paid is defined, Your Application will write this value in a Windows Registry variable. The AMI Engine continuously monitoring the Windows Registry variables will detect that the value to charge, will activate the Bill Validator and/or Change Acceptor and wait for the user to complete the payment.

While the AMI Engine is receiving the payment from the user it will continue updating the Windows Registry variables with the due amount, the status of the equipments and the credit already paid. At the same time, your application will wait and monitor the Windows Registry variables until the payment is done, before continuing with the next step in the selling process.

You can also setup in Your Application a "Cancel" button, so the user is able to cancel the payment process at any time. This is also done using the Windows Registry variables communicating with the AMI Engine.

Once the payment is done the AMI Engine will calculate the difference between the due amount and the credit, if your Coin Acceptor is able to dispense change and that function is enabled in the AMI Engine Settings, the user will receive his change in coins.

2.3 Software License

Every APS Master Interface Hardware is sold with one license of the APS Master Interface Engine software. If you want to install the AMI Engine in a different computer from the one that the AMI will be used, you need to request a new license to APS.

If you have any questions regarding our License Policy please contact us (Use contact information provided in Section 1.3 of this Manual).

2.4 System Requirements

Before you install your APS Master Interface Engine software, please make sure your computer meets the following minimal system requirements:

- Microsoft(R) Windows XP(R) Professional with SP2
- Intel(R) Pentium(R) Processor (1.73 GHZ)
- 1 LAN Network Port
- 1 Universal Serial Bus (USB 2.0)

2.5 Hardware and Software Compatibility

As we continue looking to increase the amount of Bill Validators and Change Acceptors tested with our APS Master Interface we can say that most of the MDB protocol devices are compatible with our equipment with minor adjustments.

With the development of the APS Master Interface Engine now you are able to configure each Brand and Model according to the factory settings preloaded in every equipment. If for any reason you need to change the Brand of the equipment used, you can easily do it using the AMI Engine Console to update your settings.

Here is the list of the equipment successfully tested by APS:

 <p>Change Acceptor Brand: MEI Model: Cashflow 7000 Protocol: MDB</p>	 <p>Change Acceptor Brand: MEI Model: Cashflow 690/691 Protocol: MDB</p>	 <p>Change Acceptor Brand: COINCO Model: GLOBAL MXPF703 Protocol: MDB</p>
 <p>Bill Validator Brand: CashCode Model: Backload Validator Protocol: MDB</p>	 <p>Bill Validator Brand: ICT Model: TAO V MXP4 Protocol: MDB</p>	

If you have a different equipment not listed in this table, please feel free to contact our Technical Support Staff to get the latest information on tested equipment (Use contact information provided in Section 1.3 of this Manual).

3.0 GETTING STARTED

3.1 Extracting from the .zip files

The APS Master Interface hardware requires USB drivers to work with your computer, those drivers are distributed in a .zip file (APS_Master_Interface_Driver.zip) containing 2 files that you will need once the Interface is connected for the first time to your computer's USB port (Details on how to install the Interface will be described in the next chapters).

Nombre	Tamaño	Tipo	Fecha de modificación
APS_MI.inf	2 KB	Información sobre l...	03/19/2009 22:22
APS_MI.sys	17 KB	Archivo de sistema	09/05/2001 09:50

The SDK for APS Master Interface is distributed in a .zip file (SDK_APS_Master_Interface.zip) containing all the software and drivers you will need to develop your applications. After you have uncompressed it you will find the following files:

Nombre	Tamaño	Tipo	Fecha de modificación
APS_Master_Interface_Engine.exe	160 KB	Aplicación	04/09/2009 18:09
APSMaster.dll	252 KB	Extensión de la apli...	12/16/2008 11:17
Main.frm	6 KB	Visual Basic Form File	04/09/2009 17:12
MSSCCPRJ.SCC	1 KB	Microsoft SourceSaf...	04/09/2009 00:59
SDK_APS_Master_Interface.exe	24 KB	Aplicación	04/09/2009 17:11
SDK_APS_Master_Interface.vbp	1 KB	Visual Basic Project	04/09/2009 17:12
SDK_APS_Master_Interface.vbw	1 KB	Visual Basic Project ...	04/09/2009 17:12

3.1.2 APS_Master_Interface_Engine.exe

APS_Master_Interface_Engine.exe is the main application that will be used to setup the parameters and also is used to run in the background to communicate with your application and control the functions of the Bill Validator and/or Coin Acceptor.

3.1.3 APSMaster.dll

APSMaster.dll is the "Dynamic Link Library" that contains all the functions required by AMI Engine to control the Bill Validator and/or Coin Acceptor.

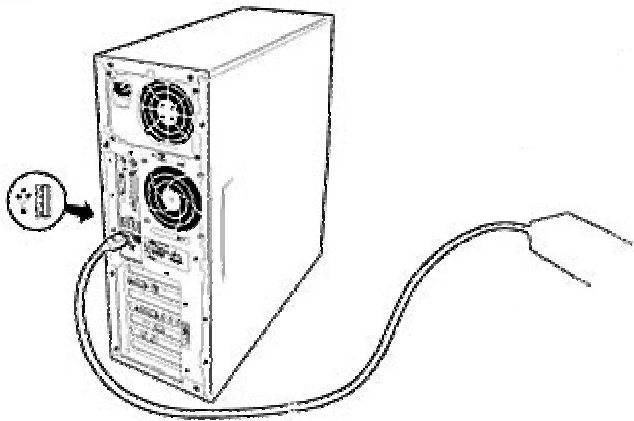
3.1.4 SDK_APS_Master_Interface.vbp

SDK_APS_Master_Interface.vbp is a sample application written in Microsoft(R) Visual Basic 6.0 to demonstrate the use of the AMI Engine. Feel free to modify the source code to integrate it to your application.

All other files in the uncompressed folder are related to this sample Visual Basic Project.

3.2 Installing the APS Master Interface

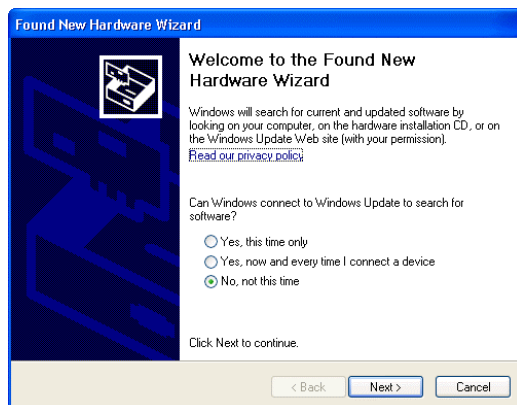
After you uncompressed the drivers in your computer follow the next steps to Install your APS Master Interface for the first time:



1. Connect the APS Master Interface to your computer using an USB Cable.

Note: At this stage it is not necessary to connect the MDB Master Cable to your Bill Validator and/or Coin Acceptor.

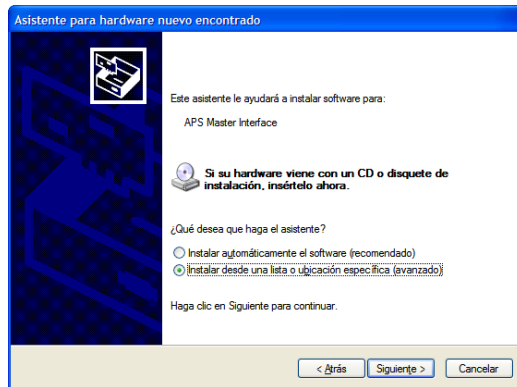
The APS Master Interface hardware is powered up directly from the USB port so it is not necessary to connect the power supply neither.



2. Windows XP will detect your new hardware and start the "Found New Hardware Wizard".

Select the "No, not this time" option.

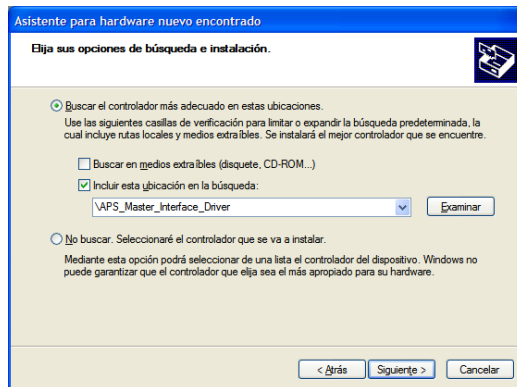
Click the "Next >" button.



3. The wizard will show the name of the "APS Master Interface" and ask if you have the installation files.

Select the "Install from a list or specific location (Advanced)" option.

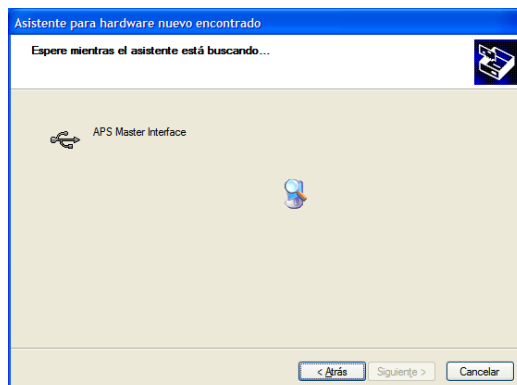
Click the "Next >" button.



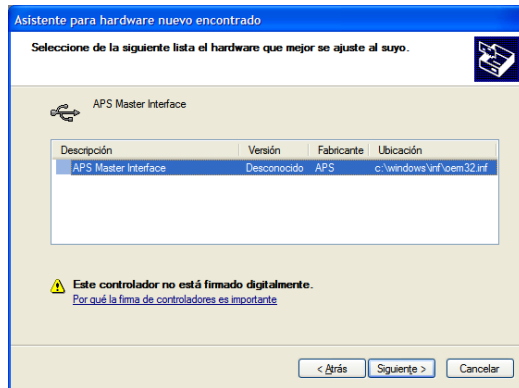
4. The wizard will ask you to choose your search and installation options.

Select "Search for the best drivers in this locations." and disable "Search removable media (floppy, CD-ROM...)".

Select "Include this location in the search:" then click the "Browse" button to find the folder "APS_Master_Interface_Driver".



5. The wizard will search for the APS Master Interface drivers in the selected folder.



6. If the destination folder was selected correctly the wizard will show the "APS Master Interface" driver.

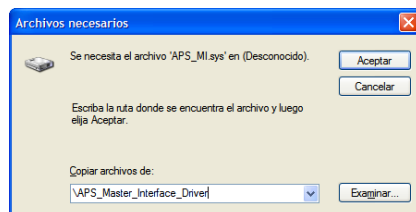
Select the "APS Master Interface".

Click the "Next >" button.



7. You will receive a notification stating that the APS Master Interface "has not passed Windows Logo testing to verify its compatibility with Windows XP".

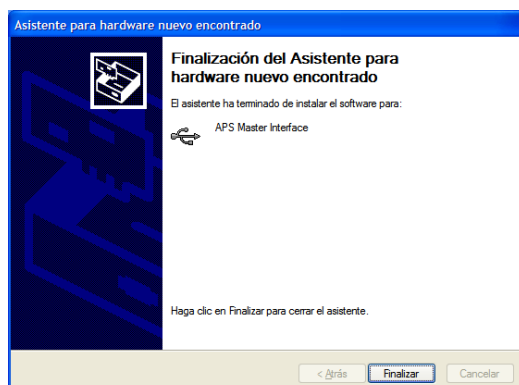
Ignore the message and click the "Continue Anyway" button.



8. The wizard will ask you to find the "APS_MI.sys" file.

Click on the "Browse" button, find the "APS_Master_Interface_Driver" folder then select the "APS_MI.sys" file.

Click "Ok".



9. Congratulations! now your APS Master Interface is installed and ready to use.

Click "Finish".

3.3 Installing the APS Master Interface Engine

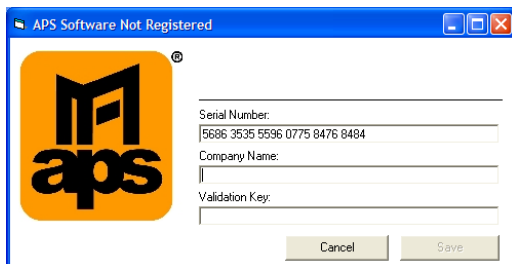
As we explained before the APS Master Interface Engine is the main software that will be used to setup the parameters of the Bill Validator and/or Coin Acceptor. Once your application is operating, the AMI Engine will be running in the background to communicate with your application and control the equipment connected to the interface.



IMPORTANT: The AMI Engine license can be used in only one computer. If you need to install the AMI Engine in a different computer you need to buy a new license (Please refer to chapter 2.3 for more details on this matter). Once the software is installed, you cannot uninstall and install it again in a new computer.

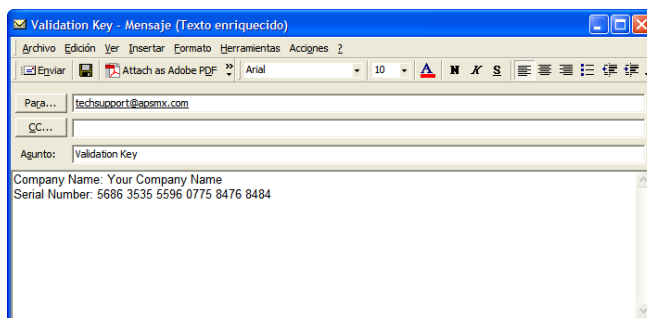
Nombre	Tamaño	Tipo	Fecha de modificación
APS_Master_Interface_Engine.exe	160 KB	Aplicación	04/09/2009 18:09
APSMaster.dll	252 KB	Extensión de la apl...	12/16/2008 11:17
Main.frm	6 KB	Visual Basic Form File	04/09/2009 17:12
MSSCCPRJ.SCC	1 KB	Microsoft SourceSaf...	04/09/2009 00:59
SDK_APS_Master_Interface.exe	24 KB	Aplicación	04/09/2009 17:11
SDK_APS_Master_Interface.vbp	1 KB	Visual Basic Project	04/09/2009 17:12
SDK_APS_Master_Interface.vbw	1 KB	Visual Basic Project ...	04/09/2009 17:12

1. Find the "APS_Master_Interface_Engine.exe" file and double click on it.



2. You will receive a warning "APS Software Not Registered".

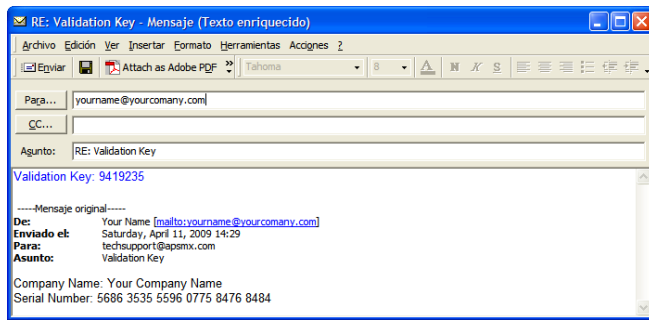
The AMI Engine software will generate an unique Serial Number for your computer.



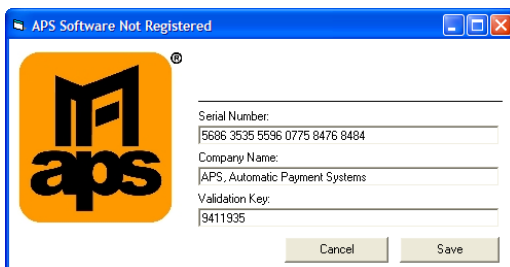
3. Send an email to APS requesting your Validation Key (email address can be found in chapter 1.3).

Write the name of your Company.

Copy and paste the Serial Number into the body of your message.

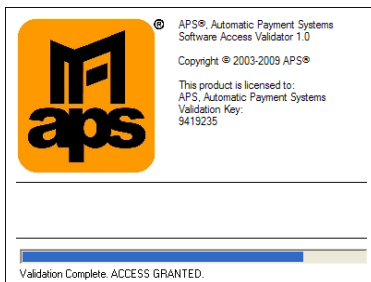


4. You will receive an email back from APS with your Validation Key number.



5. Once you get the Validation Key you will need to open the AMI Engine again, once you get to the "APS Software Not Registered" screen write your Company Name and Validation Key in the appropriate text boxes.

Click the "Save" button (The screen will be closed).



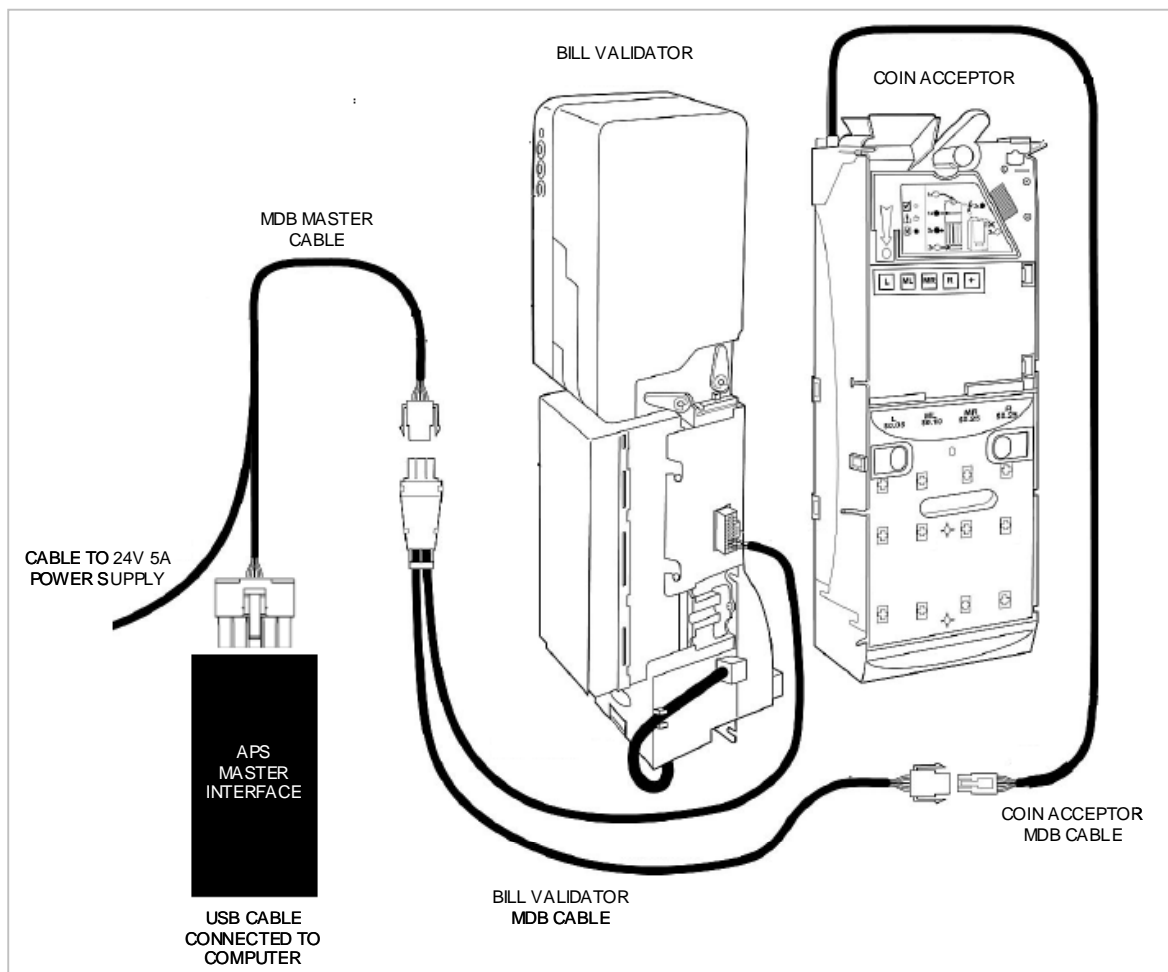
6. In the next chapters you will learn that there are two methods of opening the AMI Engine. One is by double clicking in the .exe file and the second is calling it directly from the Command prompt (used to run the application in the background).

Regardless of the method you use, if you Registered your software correctly, you will see this screen every time you run the AMI Engine application.

3.4 Connecting the Bill Validator and/or Coin Acceptor

Once you have installed your APS Master Interface Engine software and registered it you are ready to connect the Bill Validator and/or Coin Acceptor to you APS Master Interface. As we mentioned before, the AMI will get the power from the USB connection. You will need a Power Supply for the Bill Validator and/or Coin Acceptor (Power requirements change between different brands, usually a 24V 5A power supply should work with your MDB devices).

Please follow the picture bellow as a guide to connect your equipments to the AMI. You can use a stand alone Coin Acceptor or Bill Validator. In case you want to use both equipments, first you must connect your Bill Validator to the MDB Master Cable and then use the built in cable to connect your Coin Acceptor as it is shown in the picture bellow.



Some equipments have to be configured to support the MDB protocol. Please make sure your equipment meet this requirement before connecting it to the AMI. If your equipment is already included in the list of equipment tested by APS (Chapter 2.5) and you need additional information on how to set it up, please feel free to contact us to support you on this task.

4.0 USING THE SOFTWARE

The APS Master Interface Engine software has two operating modes: APS Master Interface Engine Console (Used to setup parameters, debug your application and test Bill Validator and/or Coin Acceptor) and the APS Master Interface Engine Hide (Used to control the Bill Validator and/or Coin Acceptor while your application is running. In this mode the AMI Engine won't be visible to the user).

4.1 APS Master Interface Engine Console

Before starting the APS Master Interface Engine Console make sure your AMI Interface and Bill Validator and/or Coin Acceptor are connected and powered.

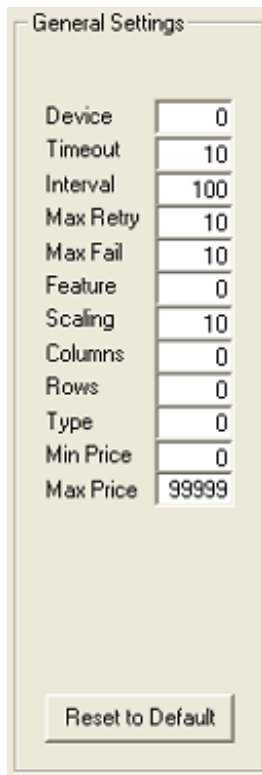
To start the AMI Engine Console you just need to double click on the file APS_Master_Interface_Engine.exe and you will see the following screen:

APS Master Interface Engine Console

General Settings		Bill Validator Settings		Coin Acceptor Settings		Test Console
Device	0	<input checked="" type="checkbox"/> Enabled		<input checked="" type="checkbox"/> Enabled		
Timeout	10	#01 <input checked="" type="checkbox"/>	20	#01 <input checked="" type="checkbox"/>	N/A	<div><div>\$0.00</div><div>Amount <input type="text" value="0"/></div><div>Pay</div></div>
Interval	100	#02 <input checked="" type="checkbox"/>	50	#02 <input checked="" type="checkbox"/>	N/A	
Max Retry	10	#03 <input checked="" type="checkbox"/>	100	#03 <input checked="" type="checkbox"/>	N/A	
Max Fail	10	#04 <input checked="" type="checkbox"/>	200	#04 <input checked="" type="checkbox"/>	N/A	
Feature	0	#05 <input checked="" type="checkbox"/>	500	#05 <input checked="" type="checkbox"/>	N/A	
Scaling	10	#06 <input checked="" type="checkbox"/>	1000	#06 <input checked="" type="checkbox"/>	N/A	
Columns	0	#07 <input type="checkbox"/>	0	#07 <input checked="" type="checkbox"/>	N/A	
Rows	0	#08 <input type="checkbox"/>	0	#08 <input checked="" type="checkbox"/>	N/A	
Type	0	#09 <input type="checkbox"/>	0	#09 <input checked="" type="checkbox"/>	N/A	
Min Price	0	#10 <input type="checkbox"/>	0	#10 <input checked="" type="checkbox"/>	N/A	
Max Price	99999	#11 <input type="checkbox"/>	0	#11 <input type="checkbox"/>	N/A	
		#12 <input type="checkbox"/>	0	#12 <input type="checkbox"/>	N/A	
		#13 <input type="checkbox"/>	0	#13 <input type="checkbox"/>	N/A	
		#14 <input type="checkbox"/>	0	#14 <input type="checkbox"/>	N/A	
		#15 <input type="checkbox"/>	0	#15 <input type="checkbox"/>	N/A	
		#16 <input type="checkbox"/>	0	#16 <input type="checkbox"/>	N/A	
<input type="button" value="Reset to Default"/>		<input type="button" value="Reset to Default"/>		<input checked="" type="checkbox"/> Dispense Change <input type="button" value="Reset to Default"/>		
APS, Automatic Payment Systems				<input type="button" value="Cancel"/> <input type="button" value="Save"/>		

The AMI Engine Console is divided in four sections: General Settings, Bill Validator Settings, Coin Acceptor Settings and Test Console. In the following sections we will describe in detail every single one.

4.1.1 General Settings



The image shows a 'General Settings' dialog box with a list of parameters and their values. At the bottom is a 'Reset to Default' button.

Parameter	Value
Device	0
Timeout	10
Interval	100
Max Retry	10
Max Fail	10
Feature	0
Scaling	10
Columns	0
Rows	0
Type	0
Min Price	0
Max Price	99999

Reset to Default

General Settings are variables used to initialize the APS Master Interface, these settings define the communication parameters between the computer and the AMI.

Normally there is not need to modify the values on the General Settings, these are predefined by APS. If you computer meets the specification described in chapter 2.4 then you don't need to change them.

If you have a computer that doesn't meet wit the specification described in chapter 2.4, please contact us to help you to define the best parameters for your system (Please note that not all the systems will work with our AMI).

The most common applications are developed to control one Bill Validator and/or one Coin Acceptor using one AMI only. Since our AMI Engine is designed to work with your USB ports, we can develop special applications to have one computer controlling more than one AMI (If this is your case, please contact us to evaluate the feasibility of your project).

If you make changes to the General Settings you need to press the "Save" button so they are available next time you run the AMI Engine Console (When you click "Save" the AMI Engine application will be closed).

4.1.2 Bill Validator Settings

Bill Type	Enabled	Value
#01	<input checked="" type="checkbox"/>	20
#02	<input checked="" type="checkbox"/>	50
#03	<input checked="" type="checkbox"/>	100
#04	<input checked="" type="checkbox"/>	200
#05	<input checked="" type="checkbox"/>	500
#06	<input checked="" type="checkbox"/>	1000
#07	<input type="checkbox"/>	0
#08	<input type="checkbox"/>	0
#09	<input type="checkbox"/>	0
#10	<input type="checkbox"/>	0
#11	<input type="checkbox"/>	0
#12	<input type="checkbox"/>	0
#13	<input type="checkbox"/>	0
#14	<input type="checkbox"/>	0
#15	<input type="checkbox"/>	0
#16	<input type="checkbox"/>	0

Reset to Default

Bill Validator Settings are used to enable/disable the use of a Bill Validator and also to specify the value of a bill assigned to each type.

Click on the Enabled checkbox to activate/deactivate the Bill Validator.

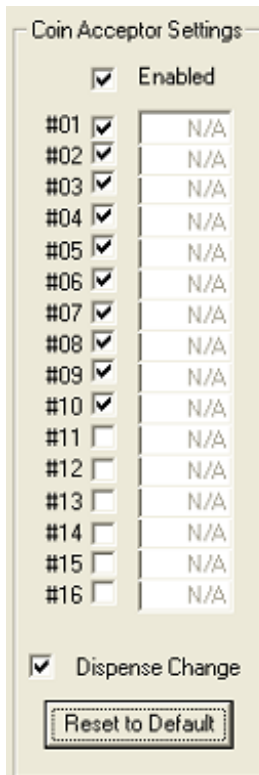
MDB protocol requires Bill Validators to send the "Bill Type" inserted by the user when the equipment is operating. There are 16 different types of bill supported by the MDB protocol.

The "Bill Type" is a number from 1 to 16. Each brand has different assignments for the value of the "Bill Type".

Default values for the Bill Validator Settings were defined using a CashCode Backload Bill Validator where the "Bill Type" number 1 has a value of 20 pesos, number 2 is 50 pesos, etc... "Bill Type" from number 7 to number 16 are not used in this type of Bill Validator.

You can disable the use of a "Bill Type" by clicking in the checkbox next to it. Don't forget to click "Save" if you want to keep your changes.

4.1.3 Coin Acceptor Settings



The image shows a software window titled "Coin Acceptor Settings". At the top, there is a checkbox labeled "Enabled" which is checked. Below this is a list of 16 coin types, numbered #01 through #16. Each number has a checkbox next to it and a corresponding value in a text box. Coins #01 through #10 have their checkboxes checked and their values are "N/A". Coins #11 through #16 have their checkboxes unchecked and their values are "N/A". At the bottom of the window, there is a checkbox labeled "Dispense Change" which is checked, and a button labeled "Reset to Default".

Coin Type	Enabled	Value
#01	<input checked="" type="checkbox"/>	N/A
#02	<input checked="" type="checkbox"/>	N/A
#03	<input checked="" type="checkbox"/>	N/A
#04	<input checked="" type="checkbox"/>	N/A
#05	<input checked="" type="checkbox"/>	N/A
#06	<input checked="" type="checkbox"/>	N/A
#07	<input checked="" type="checkbox"/>	N/A
#08	<input checked="" type="checkbox"/>	N/A
#09	<input checked="" type="checkbox"/>	N/A
#10	<input checked="" type="checkbox"/>	N/A
#11	<input type="checkbox"/>	N/A
#12	<input type="checkbox"/>	N/A
#13	<input type="checkbox"/>	N/A
#14	<input type="checkbox"/>	N/A
#15	<input type="checkbox"/>	N/A
#16	<input type="checkbox"/>	N/A

☒ Dispense Change

Reset to Default

Coin Acceptor Settings are used to enable/disable the use of a Coin Acceptor and also to specify the type of coins to be used. Please note that unlike the Bill Validator in the Coin Acceptor it is not possible to assign the value that each type of coin. This information is already programmed in the Coin Acceptor and will be communicated to the AMI Engine once the equipment is activated.

Some Coin Acceptors manufacturers use different Scales when they setup the equipments. If you notice that the value of the coins inserted doesn't match with the one returned by the AMI Engine Console, you can adjust the "Scaling" factor in the General Settings (Chapter 4.1.1).

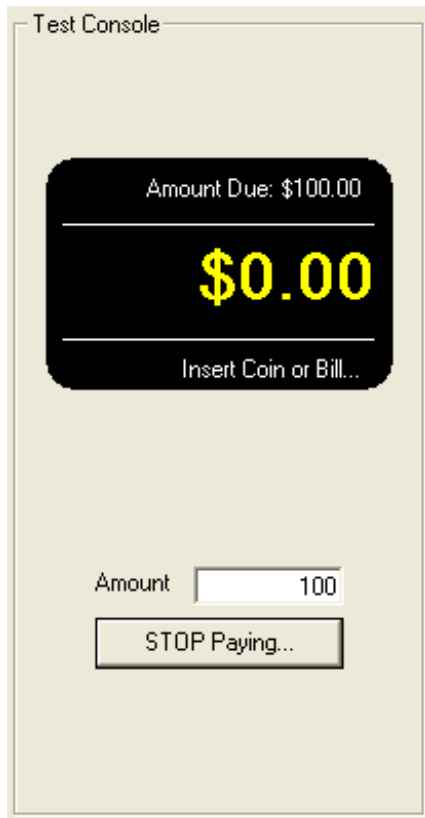
To enable/disable the use of a coin, just click on the checkbox next to the "Coin Type" number.

With this Version of the AMI Engine Console it is not possible to get the actual value of the coins programmed in the Coin Acceptors, but we already left a column that will be used for this proposes in the near future (currently is showing the "N/A" value).

If you want to dispense change using your Coin Acceptor you just need to activate the function by clicking in the "Dispense Change" checkbox.

Please note that the availability of dispensing change is limited by the quantity and type of coins that you Coin Acceptor will hold on the tubes when this function is required. We highly recommend the use of additional "Hoppers" when a high amount of transactions will require dispensing change.

4.1.4 Test Console



Once you have finished with the setup of your parameters you can use the Test Console to verify that your Bill Validator and/or Coin Acceptor are working correctly.

If you made changes to the parameter, it is necessary to click the "Save" button, this will automatically close the AMI Engine Console application. Once you open again the AMI Engine Console the Bill Validator and/or Coin Acceptor will receive the new settings defined in your parameters.

To test your equipment you just need to type the Amount of money to be charged in the "Amount" textbox, then click the "Pay" button.

The AMI Engine Console will activate the Bill Validator and/or Coin Acceptor so you can start inserting the money.

When you click the "Pay" button, the function for that button will change to "STOP Paying...". You can click on this button at any time to cancel the paying process.

Once the payment operation is in process, the Status Display will change to an active role displaying the status and balance of the amount due. The Status Display is divided in three sections.

- a. The top section displays the amount of money due.
- b. The middle section displays the credit or amount of money already paid.
- c. The bottom section displays the status of the AMI Engine Status. There are six types of status that the AMI Engine uses: Bill Inserted, Coin Inserted, Processing Payment, Insert Coin or Bill, Dispensing Change and Change Dispensed.

AMI Engine Test Console is very useful when you are writing your applications, you can always use it to debug your code and monitor the payment functions. When you integrate AMI Engine to your application, this is the same software that will communicate with your devices, but the AMI Engine will be hidden to the user.

4.2 APS Master Interface Engine Console "Hide"

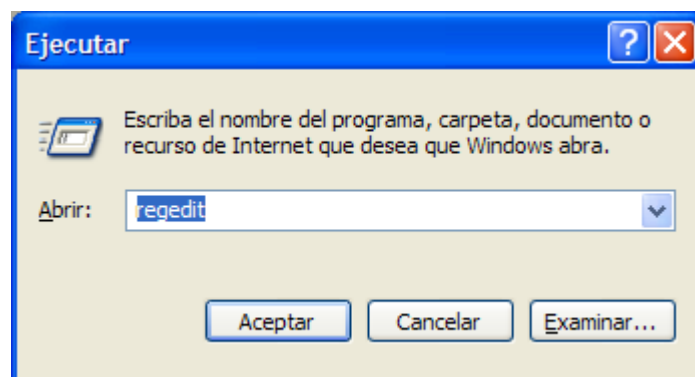
The APS Master Interface Engine Console "Hide" is the same application described in the previous chapter, the only difference is that the application runs in the background without displaying any Forms. While you are using the AMI Engine Console using the "Hide" parameter, you will be able to control the payment process using your Bill Validator and/or Coin Acceptor, but you won't be able to change the parameters.

4.2.1 Windows Registry Settings for AMI Engine Console

The Windows Registry is a database which stores settings and options for Microsoft Windows operating systems. It contains information and settings for hardware, operating system software, most non-operating system software, and per-user settings. The registry also provides a window into the operation of the kernel, exposing runtime information such as performance counters and currently active hardware.

AMI Engine Console uses the Windows Registry database to save the application parameters and also to communicate with you application. By using this database we are able to support any programming language since the values are available from any platform.

Windows Registry can be accessed manually using the "Regedit" utility from Windows XP.

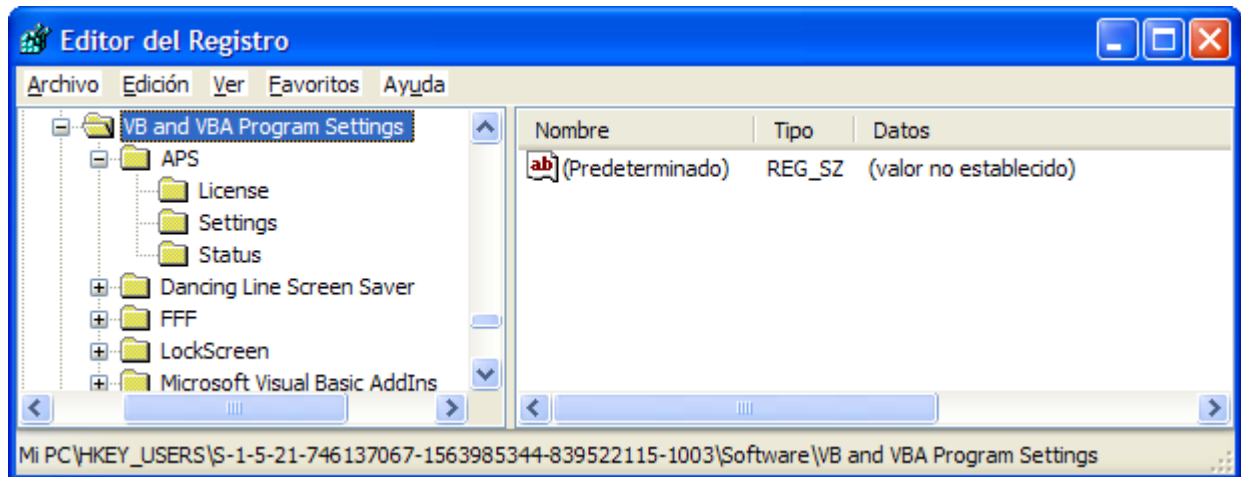


Once the Windows Registry editor is open, you can find the AMI Engine Console parameters under the HKEY_USERS directory in My PC:

My PC\HKEY_USERS\S???\\Software\\VB and VBA Program Settings\\APS

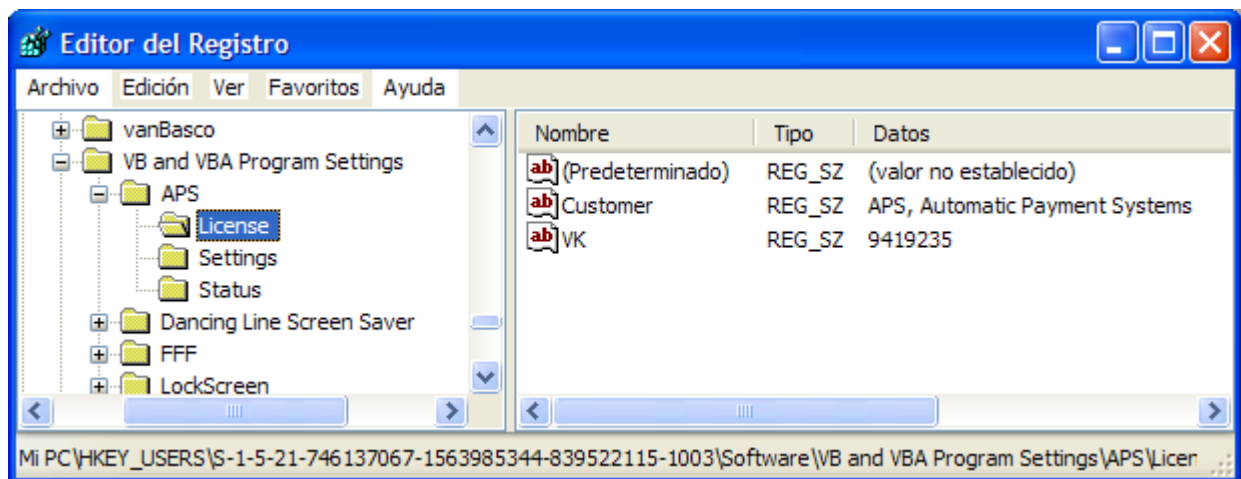
Note: As the "S???" can be different from one computer to another, you can also search for the right path using the "Search" menu option typing "VB and VBA Program Settings" as the searched item.

AMI Engine Console uses three Windows Registry categories to storage all the parameters and variables to communicate with your applications: License, Settings and Status.



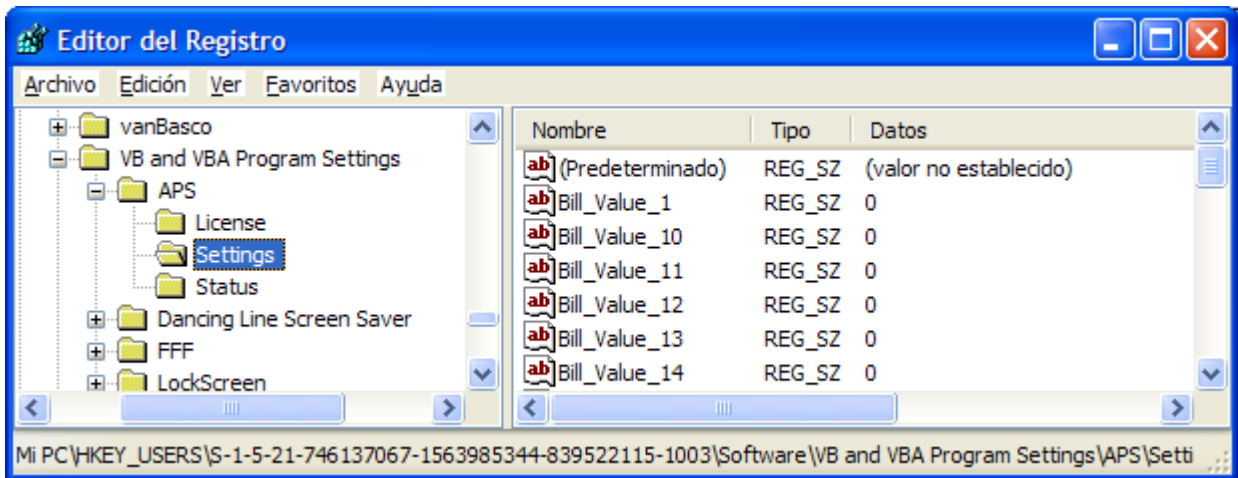
4.2.2 License Windows Registry

License Windows Registry is used to record the values used by the AMI Engine Console validation software to authenticate the license number registered by the customer. As this value can be changed manually we don't recommend it because any mistake on typing the numbers can block you from using the AMI Engine Console application.



4.2.3 Settings Windows Registry

Settings Windows Registry is used to record the values used by the AMI Engine Console to set the parameters of the Bill Validator and Coin Acceptor. As this values can be changed manually we recommend using the AMI Engine Console described in Chapter 4.1



4.2.3 Status Windows Registry

Status Windows Registry values are used to communicate the AMI Engine Console with your application. While there are five variables under this section, only three (Due, Cancel, Exit) should be updated by your application and the other two (Credit, Status) will be automatically updated by the AMI Engine Console.

